

[illegible]

```

CCCCCCCC HH      HH RRRRRRRR SSSSSSSS UU      UU 00000000
CCCCCCCC HH      HH RRRRRRRR SSSSSSSS UU      UU 00000000
CC        HH      HH RR        RR SS      SS UU      UU 00      00
CC        HH      HH RR        RR SS      SS UU      UU 00      00
CC        HH      HH RR        RR SS      SS UU      UU 00      00
CC        HH      HH RR        RR SS      SS UU      UU 00      00
CC        HHHHHHHHHH RRRRRRRR SSSSSS UU      UU 00000000
CC        HHHHHHHHHH RRRRRRRR SSSSSS UU      UU 00000000
CC        HH      HH RR      RR SS      SS UU      UU 00      00
CC        HH      HH RR      RR SS      SS UU      UU 00      00
CC        HH      HH RR      RR SS      SS UU      UU 00      00
CC        HH      HH RR      RR SS      SS UU      UU 00      00
CCCCCCCC HH      HH RR      RR SSSSSSSS UUUUUUUUUU 00000000
CCCCCCCC HH      HH RR      RR SSSSSSSS UUUUUUUUUU 00000000

```

....  
....  
....  
....

```

LL          IIIIII SSSSSSSS
LL          IIIIII SSSSSSSS
LL          II     SS
LL          II     SS
LL          II     SS
LL          II     SS
LL          II     SSSSSS
LL          II     SSSSSS
LL          II     SS
LL          II     SS
LL          II     SS
LL          II     SS
LLLLLLLLLL IIIIII SSSSSSSS
LLLLLLLLLL IIIIII SSSSSSSS

```

(2)	44	DECLARATIONS
(3)	97	TEST A CHARACTER FOR CLASS
(5)	185	GET TOKEN
(6)	242	SET NONE BLANK

```
0000 1      .TITLE  CHRSUB - CHARACTER MANIPULATION SUBROUTINES
0000 2      .IDENT  'V04-000'
0000 3
0000 4
0000 5 *****
0000 6 *****
0000 7      *  COPYRIGHT (c) 1978, 1980, 1982, 1984 BY
0000 8      *  DIGITAL EQUIPMENT CORPORATION, MAYNARD, MASSACHUSETTS.
0000 9      *  ALL RIGHTS RESERVED.
0000 10
0000 11      *  THIS SOFTWARE IS FURNISHED UNDER A LICENSE AND MAY BE USED AND COPIED
0000 12      *  ONLY IN ACCORDANCE WITH THE TERMS OF SUCH LICENSE AND WITH THE
0000 13      *  INCLUSION OF THE ABOVE COPYRIGHT NOTICE. THIS SOFTWARE OR ANY OTHER
0000 14      *  COPIES THEREOF MAY NOT BE PROVIDED OR OTHERWISE MADE AVAILABLE TO ANY
0000 15      *  OTHER PERSON. NO TITLE TO AND OWNERSHIP OF THE SOFTWARE IS HEREBY
0000 16      *  TRANSFERRED.
0000 17
0000 18      *  THE INFORMATION IN THIS SOFTWARE IS SUBJECT TO CHANGE WITHOUT NOTICE
0000 19      *  AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT
0000 20      *  CORPORATION.
0000 21
0000 22      *  DIGITAL ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS
0000 23      *  SOFTWARE ON EQUIPMENT WHICH IS NOT SUPPLIED BY DIGITAL.
0000 24
0000 25 *****
0000 26 *****
0000 27
0000 28
0000 29      ++
0000 30      FACILITY:      UTILITY SUBROUTINES
0000 31
0000 32      ABSTRACT:      CHARACTER MANIPULATION SUBROUTINES
0000 33
0000 34      ENVIRONMENT:    NATIVE/USER MODE CODE
0000 35
0000 36      AUTHOR:         W.H.BROWN, CREATION DATE:      19-MAY-1977
0000 37
0000 38      MODIFIED BY:
0000 39
0000 40      : : VERSION
0000 41      01  -
0000 42      --
```



```
0000 44      .SBTTL  DECLARATIONS
0000 45      :
0000 46      : INCLUDE FILES:
0000 47      :
0000 48      :
0000 49      :
0000 50      : MACROS:
0000 51      :
0000 52      : MACRO TO GENERATE AN ENTRY IN THE CHARACTER CLASSIFICATION TABLE
0000 53      :
0000 54      : CALL:
0000 55      : CHAR    NAME,CHR
0000 56      : WHERE:
0000 57      : NAME IS THE SYMBOLIC NAME SUFFIX TO 'CHR$K_' FOR THE CHAR
0000 58      : CHR IS THE ASCII CHAR.
0000 59      :
0000 60      : .MACRO CHAR    NAME,CHR,N
0000 61      : CHR$K_'NAME == N
0000 62      : .BYTE ^A\CHR\
0000 63      : .ENDM
0000 64      :
0000 65      :
0000 66      : EQUATED SYMBOLS:
0000 67      :
0000 68      : DEFINE SPECIAL SYMBOLS FOR ALPHA/NUMERIC SETS
0000 69      :
00000001 0000 70      : CHR$K_ALPHA == 1
00000002 0000 71      : CHR$K_NUMERIC == 2
0000 72      :
0000 73      :
0000 74      : OWN STORAGE:
0000 75      :
00000000 76      : .PSECT  _PURE    RD,NOWRT,BYTE
0000 77      :
0000 78      : CHRTBL:
0000 79      : CHAR    SLASH    </>    12
0001 80      : CHAR    SEMI    <:>    11
0002 81      : CHAR    LBRACK <[>    10
0003 82      : CHAR    RBRACK <]>    9
0004 83      : CHAR    COMMA   <,>    8
0005 84      : CHAR    DOT     <.>    7
0006 85      : CHAR    COLON   <:>    6
0007 86      : CHAR    BLANK   < >    5
0008 87      : CHAR    DOLLAR <$>    4
0009 88      : CHAR    UNDRSCR <_>    3
00 00 000A 89      : .BYTE    0,0
000C 90      :
0000000C 000C 91      : CHRTBLSIZ = . - CHRTBL
000C 92      :
2B 25 2D 000C 93      : SPCNUM: .ASCII \-Z+\
00000003 000F 94      : SPCNUMSIZ = . - SPCNUM
000F 95      :
```

; EOL AND FILLER FOR REMAINING COUNT

; SPECIAL CHARACTERS TREATED AS NUMERIC

```
000F 97 .SBTTL TEST A CHARACTER FOR CLASS
000F 98 :++
000F 99 : FUNCTIONAL DESCRIPTION:
000F 100 :
000F 101 : THIS ROUTINE IS CALLED TO CLASSIFY AN ASCII CHARATER INTO
000F 102 : ONE OF SEVERAL CLASSES. AN ALTERNATE ENTRY PROVIDES LOWER
000F 103 : TO UPPER CASE CONVERSION AS WELL.
000F 104 :
000F 105 : CALLING SEQUENCE:
000F 106 :
000F 107 : BSB/JSB CHR$TSTCHR : TEST THE CHARACTER
000F 108 : BSB/JSB CHR$CVT : CONVERT AND TEST
000F 109 :
000F 110 : INPUT PARAMETERS:
000F 111 :
000F 112 : R6 CONTAINS ADDRESS OF BYTE TO TEST
000F 113 :
000F 114 : IMPLICIT INPUTS:
000F 115 :
000F 116 : STRING IS TERMINATED BY A ZERO BYTE
000F 117 :
000F 118 : OUTPUT PARAMETERS:
000F 119 :
000F 120 : R0 SET TO "CHR$K <CLASS_NAME>" IF ONE OF RECOGNIZED CHARACTERS
000F 121 : ELSE SET TO MINUS 1
000F 122 :
000F 123 : IMPLICIT OUTPUTS:
000F 124 :
000F 125 : NONE
000F 126 :
000F 127 : COMPLETION CODES:
000F 128 :
000F 129 : NONE
000F 130 :
000F 131 : SIDE EFFECTS:
000F 132 :
000F 133 : NONE
000F 134 :
000F 135 : --
000F 136 :
000F 137 CHR$CVT:: : CONVERT TO UPPER CASE
61 8F 66 91 000F 138 CMPB (R6),#<^A/A/+^X20> : LOWER CASE A?
0D 19 0013 139 BLSS CHR$TSTCHR : BR IF NOT LOWER
7A 8F 66 91 0015 140 CMPB (R6),#<^A/Z/+^X20> : LOWER CASE Z?
07 14 0019 141 BGTR CHR$TSTCHR : BR IF NOT LOWER
66 20 82 001B 142 SUBB #^X20,(R6) : CONVERT TO UPPER
02 11 001E 143 BRB CHR$TSTCHR :
0020 144 :
0020 145 CHR$TSTNXT:: : TEST NEXT CHAR
56 D6 0020 146 INCL R6 : ADD ONE T1 ADDRESS
0022 147 :
0022 148 CHR$TSTCHR:: : TEST A CHARACTER FOR CLASS
50 D4 0022 149 CLRL R0 : ASSUME END-OF-LINE
66 95 0024 150 TSTB (R6) : END-OF-LINE?
50 13 0026 151 BEQL 90$ : BR IF YES
50 D6 0028 152 INCL R0 : SET TYPE TO ALPHA
41 8F 66 91 002A 153 CMPB (R6),#^A/A/ : CHECK AGAINST LOW LIMIT
```

		12	1F	002E	154	BLSSU	20\$	:	BR IF BELOW ALPHA
5A 8F	66	91	0030	155	CMPB	(R6),#^A/Z/	:	NOW CHECK HI END	
	42	15	0034	156	BLEQ	90\$	:	BR IF ALPHA	
61 8F	66	91	0036	157	CMPB	(R6),#<^A/A/+^X20>	:	CHECK FOR LOWER CASE ALPHA	
	06	19	003A	158	BLSS	20\$	:	BR IF NO	
7A 8F	66	91	003C	159	CMPB	(R6),#<^A/Z/+^X20>	:	OTHER LIMIT	
	36	15	0040	160	BLEQ	90\$	:	FOUND THE CLASS	
	02	DD	0042	161	PUSHL	S^#CHRSK NUMERIC	:	SET VALUE FOR NUMERIC CHARATERS	
C3 AF	03	66	3A	0044	162	LOCC	(R6),#SPCNUMSIZ,SPCNUM	:	CHECK FOR SPECIAL NUMERIC CHARACTERS
	01	BA	0049	163	POPR	#^M<R0>	:	GET VALUE FOR NUMERIC CHARACTER	
	2B	12	004B	164	BNEQ	90\$	:	BR IF CHARACTER IS SPECIAL NUMERIC	
	30	66	91	004D	165	CMPB	(R6),#^A/0/	:	CHECK LOW LIMIT
	05	19	0050	166	BLSS	30\$	:	BR IF NOT NUMERIC	
	39	66	91	0052	167	CMPB	(R6),#^A/9/	:	WHAT ABOUT THE HI LIMIT
	21	15	0055	168	BLEQ	90\$	:	BR IF NUMERIC	
A4 AF	0C	66	3A	0057	169	LOCC	(R6),#CHRTBLSIZ,CHRTBL	:	CHECK IF ONE OF SPECIALS
	1A	12	005C	170	BNEQ	90\$	:	BR IF YES	
	50	05	D0	005E	171	MOVL	#CHRSK BLANK,R0	:	ASSUME TAB
	09	66	91	0061	172	CMPB	(R6),#^A/ /	:	IS IT A TAB?
	12	13	0064	173	BEQL	90\$	:	BR IF YES	
	50	0A	D0	0066	174	MOVL	#CHRSK LBRAKT,R0	:	ASSUME LEFT BRACKET
	3C	66	91	0069	175	CMPB	(R6),#^A/</	:	IS IT THE FUNNY BRAKET?
	0A	13	006C	176	BEQL	90\$	:	BR IF YES	
	50	D6	006E	177	INCL	R0	:	CHANGE CODE TO RIGHT BRACKET	
	3E	66	91	0070	178	CMPB	(R6),#^A/>/	:	CHECK CLOSE BRAKET
	03	13	0073	179	BEQL	90\$	:	BR IF YES	
	50	01	CE	0075	180	MNEGL	#1,R0	:	SET AS GENERAL SPECIAL
	50	D5	0078	181	TSTL	R0	:	SET STATUS BASED ON VALUE	
		05	007A	182	RSB		:		



```
007B 184
007B 185      .SBTTL  GET TOKEN
007B 186 :++
007B 187 : FUNCTIONAL DESCRIPTION:
007B 188 :
007B 189 :     THIS ROUTINE IS CALLED TO PARSE THE NEXT TOKEN FROM THE
007B 190 :     COMMAND LINE.
007B 191 :
007B 192 : CALLING SEQUENCE:
007B 193 :
007B 194 :     BSB/JSB CHR$GETOKEN      : GET TOKEN FROM LINE
007B 195 :     BSB/JSB CHR$NXTOKEN     : TOKEN AFTER NEXT CHARACTER
007B 196 :
007B 197 : INPUT PARAMETERS:
007B 198 :
007B 199 :     R6 CONTAINS ADDRESS OF NEXT BYTE ON THE LINE
007B 200 :
007B 201 : IMPLICIT INPUTS:
007B 202 :
007B 203 :     STRING IS TERMINATED BY ZERO BYTE
007B 204 :
007B 205 : OUTPUT PARAMETERS:
007B 206 :
007B 207 :     R6 IS ADVANCED TO THE FIRST NONE BLANK CHARACTER AFTER THE TOKEN.
007B 208 :     R3,R4 ARE A DESCRIPTOR TO THE TOKEN
007B 209 :
007B 210 : IMPLICIT OUTPUTS:
007B 211 :
007B 212 :     "Z" BIT IS SET IF ZERO LENGTH TOKEN IS PARSED.
007B 213 :
007B 214 : COMPLETION CODES:
007B 215 :
007B 216 :     R0 IS SET TO THE TYPE OF THE CHARACTER
007B 217 :
007B 218 : SIDE EFFECTS:
007B 219 :
007B 220 :     NONE
007B 221 :
007B 222 :--
007B 223      .ENABL  LSB
007B 224
007B 225 CHR$GETOKEN::
007B 226     DECL    R6
007D 227 CHR$NXTOKEN::
007D 228     BSB     CHR$NXTNBLK
007F 229     MOVAB   (R6),R4
0082 230     DECL    R6
0084 231 10$:     MOVAB   1(R6),R3
0088 232     BSB     CHR$STNXT
008A 233     BEQL    40$
008C 234     CMPB    R0,#CHR$K_BLANK
008F 235     BL      10$
0091 236     LJE     40$
0093 237     BSB     CHR$NXTNBLK
0095 238 40$:     SUBL    R4,R3
0098 239 50$:     RSB
0098 239      GET OUT

56 D7 007B 226     DECL    R6
54 1C 10 007D 228     BSB     CHR$NXTNBLK
53 01 A6 9E 007F 229     MOVAB   (R6),R4
53 01 96 D7 0082 230     DECL    R6
05 50 9E 0084 231 10$:     MOVAB   1(R6),R3
05 F3 10 0088 232     BSB     CHR$STNXT
05 02 13 008A 233     BEQL    40$
05 06 91 008C 234     CMPB    R0,#CHR$K_BLANK
53 54 1F 008F 235     BL      10$
05 02 12 0091 236     LJE     40$
05 06 10 0093 237     BSB     CHR$NXTNBLK
53 54 C2 0095 238 40$:     SUBL    R4,R3
05 05 05 0098 239 50$:     RSB
```



```
0099 241      .DSABL  LSB
0099 242      .SBTTL  SET NONE BLANK
0099 243      :++
0099 244      : FUNCTIONAL DESCRIPTION:
0099 245      :
0099 246      : THIS ROUTINE IS CALLED TO ADVANCE THE CHARACTER POINTER
0099 247      : TO THE FIRST NONE BLANK CHARATER ON THE LINE.
0099 248      :
0099 249      : CALLING SEQUENCE:
0099 250      :
0099 251      : BSB/JSB CHR$SETNB          ; SET NONE BLANK
0099 252      :
0099 253      : INPUT PARAMETERS:
0099 254      :
0099 255      : R6 CONTAINS ADDRESS OF NEXT BYTE ON THE LINE
0099 256      :
0099 257      : IMPLICIT INPUTS:
0099 258      :
0099 259      : NONE
0099 260      :
0099 261      : OUTPUT PARAMETERS:
0099 262      :
0099 263      : R6 IS ADVANCED TO THE FIRST NONE BLANK CHARACTER
0099 264      :
0099 265      : IMPLICIT OUTPUTS:
0099 266      :
0099 267      : NONE
0099 268      :
0099 269      : COMPLETION CODES:
0099 270      :
0099 271      : R0 = 1 IF MORE DATA ON LINE, 0 IS NO NONE BLANK CHARACTERS
0099 272      :
0099 273      : SIDE EFFECTS:
0099 274      :
0099 275      : NONE
0099 276      :
0099 277      :--
0099 278      .ENABL  LSB
0099 279
0099 280 CHR$SETNBLK::          ; SET NONE BLANK
0099 281      DECL          R6          ; BACK UP SO SKIP ONE WILL BE NOP
0099 282 CHR$NXTNBLK::        ; SKIP THEN-THEN NEXT NONE BLANK
0099 283 20$: BSBW          CHR$STNXT
0099 284      BEQL          40$
0099 285      CMPB          R0,#CHR$K_BLANK
0099 286      BEQL          20$
0099 287      MOVL          #1,R0
0099 288 40$: RSB
0099 289
0099 290      .DSABL  LSB
0099 291      .END
```

56 D7 0099 281  
FF82 30 009B 282  
08 13 009E 283  
05 50 91 00A0 284  
F6 13 00A3 285  
50 01 D0 00A5 286  
05 05 00A8 287  
00A9 288  
00A9 289  
00A9 290  
00A9 291

CHRSUB  
Symbol table

- CHARACTER MANIPULATION SUBROUTINES<sup>M 4</sup>

15-SEP-1984 23:37:36 VAX/VMS Macro V04-00  
4-SEP-1984 23:15:00 [CLIUTL.SRC]CHRSUB.MAR;1

Page 7  
(6)

CHRSCVT	0000000F	RG	01
CHRSGETOKEN	0000007B	RG	01
CHRSK_ALPHA	= 00000001	G	
CHRSK_BLANK	= 00000005	G	
CHRSK_COLON	= 00000006	G	
CHRSK_COMMA	= 00000008	G	
CHRSK_DOLLAR	= 00000004	G	
CHRSK_DOT	= 00000007	G	
CHRSK_LBRAKT	= 0000000A	G	
CHRSK_NUMERIC	= 00000002	G	
CHRSK_RBRAKT	= 00000009	G	
CHRSK_SEMI	= 0000000B	G	
CHRSK_SLASH	= 0000000C	G	
CHRSK_UNDRSCR	= 00000003	G	
CHRSNXTNBLK	0000009B	RG	01
CHRSNXTOKEN	0000007D	RG	01
CHRSSETNBLK	00000099	RG	01
CHRSTSTCHR	00000022	RG	01
CHRSTSTNXT	00000020	RG	01
CHRTBL	00000000	R	01
CHRTBLSIZ	= 0000000C		
SPCNUM	0000000C	R	01
SPCNUMSIZ	= 00000003		

-----  
! Psect synopsis !  
-----

PSECT name	Allocation	PSECT No.	Attributes														
. ABS .	00000000 ( 0.)	00 ( 0.)	NOPI	USR	CON	ABS	LCL	NOSHR	NOEXE	NORD	NOWRT	NOVEC	BYTE				
_PURE	000000A9 ( 169.)	01 ( 1.)	NOPI	USR	CON	REL	LCL	NOSHR	EXE	RD	NOWRT	NOVEC	BYTE				

-----  
! Performance indicators !  
-----

Phase	Page faults	CPU Time	Elapsed Time
Initialization	12	00:00:00.12	00:00:01.49
Command processing	105	00:00:00.94	00:00:03.15
Pass 1	94	00:00:00.76	00:00:03.16
Symbol table sort	0	00:00:00.01	00:00:00.01
Pass 2	62	00:00:00.53	00:00:01.85
Symbol table output	4	00:00:00.04	00:00:00.04
Psect synopsis output	1	00:00:00.02	00:00:00.03
Cross-reference output	0	00:00:00.00	00:00:00.00
Assembler run totals	280	00:00:02.43	00:00:09.74

The working set limit was 750 pages.  
4246 bytes (9 pages) of virtual memory were used to buffer the intermediate code.  
There were 10 pages of symbol table space allocated to hold 23 non-local and 8 local symbols.  
291 source lines were read in Pass 1, producing 11 object records in Pass 2.  
1 page of virtual memory was used to define 1 macro.

-----  
! Macro library statistics !  
-----

Macro library name	Macros defined
-----	-----
\$255\$DUA28:[CLIUTL.OBJ]CLIUTL.MLB;1	0
\$255\$DUA28:[SYS.OBJ]LIB.MLB;1	0
\$255\$DUA28:[SYSLIB]STARLET.MLB;2	0
TOTALS (all libraries)	0

0 GETS were required to define 0 macros.

There were no errors, warnings or information messages.

MACRO/LIS=LISS:CHRSUB/OBJ=OBJ\$:CHRSUB MSRC\$:CHRSUB/UPDATE=(ENH\$:CHRSUB)+EXECML\$/LIB+LIB\$:CLIUTL/LIB



0049 AH-BT13A-SE  
VAX/VMS V4.0

DIGITAL EQUIPMENT CORPORATION  
CONFIDENTIAL AND PROPRIETARY

BCPRSDEF  
REQ

CVCLIB  
LIS

INFO  
LIS

TYPE  
REQ

CHRSUB  
LIS

CVCLINUM  
LIS

SHODEVDEF  
REQ

CLIMAC  
MAR

CVCLIFRM  
LIS

DIGRAMS  
LIS

CALCMAX  
LIS

CLUTLMAC  
MAR

CUTLINE  
LIS

BCMDPRS  
LIS

SHOWDEF  
REQ

CREATE  
LIS